```c
//Q1. C program to swap two variables using pointers:
#include<stdio.h>

void swap(int *x,int *y){
      int temp;
      temp = *x;
      *x = *y;
      *y = temp;
}

int main(){
      int num1,num2;
      printf("Enter num1:");
      scanf("%d", &num1);

      printf("Enter num2:");
      scanf("%d", &num2);

      printf("Before getting swapped: num1=%d, num2=%d\n", num1, num2);

      swap(&num1,&num2);

      printf("After getting swapped: num1=%d, num2=%d\n", num1, num2);

return 0;
}
```

Output:

```
Enter num1:12
Enter num2:16
Before getting swapped: num1=12, num2=16
After getting swapped: num1=16, num2=12
```

```c
//Q2. C program to read a text and count all occurances
// of a particular word.
#include<stdio.h>
#include<string.h>

int main(){
      int i=0,j=0,count = 0;
    char str[100],copy[20];
    printf("Enter a sentence:");
    gets(str);
    char word[20];
    printf("Enter the word:");
    gets(word);
    while(str[i]!='\0'){
          while(str[i]!=' '&&str[i]!='\0'){
              copy[j]=str[i];
              ++j;
              ++i;
          }
          copy[j]='\0';
          j=0;
          if((strcmp(copy,word))==0){
             count+=1;
          }
          if(str[i]=='\0')
          break;
          else
          i++;
    }
    printf("The number of occurences of %s is %d", word, count);

return 0;
}
```

Output:

```
Enter a sentence:why why why and where what why when
Enter the word:why
The number of occurences of why is 4
```

```c
//Q3. Write a C function to find the largest of three
//numbers using pass by reference.
//The function to find the largest should return a pointer.
#include<stdio.h>

int *largest(int *a, int *b, int *c){
      static int x;
      x = (*a>*b? (*a>*c?*a:*c):(*b>*c?*b:*c));
return &x;
}

int main(){
   int a,b,c;
  printf("Enter 3 numbers:\n");
  scanf("%d %d %d", &a, &b ,&c);
  printf("The largest number is %d\n", *largest(&a,&b,&c));

return 0;
}
```

Output:

```
Enter 3 numbers:
33 23 40
The largest number is 40
Saving session...
```

```c
//Q4. WACP using pointers to read in an array of
//integers and print its elements in reverse order.
#include<stdio.h>

int main(){
int arr[100];
int n,i;
int *ptr;
printf("Enter the size of the array: ");
scanf("%d", &n);
ptr = arr;
printf("Enter the elements of the array: \n");
for(i=0; i<n; i++){
      scanf("%d", ptr);
      ptr++;
}
printf("Before reversing:\n");
ptr = arr;
for(i=0; i<n; i++){
      printf("%d ", *ptr);
      ptr++;
}
printf("\nAfter reversing:\n");
ptr = &arr[n-1];
for(i=0; i<n; i++){
      printf("%d ", *ptr);
      ptr--;
}

return 0;
}
```

Output:

```
Enter the size of the array: 5
Enter the elements of the array:
1 2 3 4 5
Before reversing:
1 2 3 4 5
After reversing:
5 4 3 2 1
```

```c
//Q5. WACP to sort strings abc, deh, bac in
//ascending order using array of pointers.
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

int main(){
      char *c[3];
      int i, n=3;
      void sort(int n, char *c[]);
      for(i=0; i<n; i++){
            printf("Enter string no. %d: ", i+1);
            c[i] = (char *)malloc(3*sizeof(char));
            scanf("%s", c[i]);
            printf("\n");
      }
     sort(n,c);
      printf("\nSorted strings are:\n");
      for(i=0; i<n; i++){
            printf("%d %s\n", i+1, c[i]);
      }

return 0;
}

void sort(int n, char *c[]){
      int i,j;
      char t[20];
      for(i=0; i<n-1; i++ ){
            for(j=i+1; j<n; j++){
                  if(strcmp(c[i],c[j])>0){
                        strcpy(t,c[j]);
                        strcpy(c[j],c[i]);
                        strcpy(c[i],t);
                  }
            }
      }
return;
}
```

Output:

```
Enter string no. 1: abc

Enter string no. 2: def

Enter string no. 3: bac


Sorted strings are:
1 abc
2 bac
3 def
```